



Carrera:	Tecnatura Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

Laboratorio de Computación II

PLANIFICACIÓN

1) FUNDAMENTACIÓN DE LA MATERIA DENTRO DEL PLAN DE ESTUDIOS

Es una asignatura teórico-práctica orientada tanto a conocer aspectos complementarios en las tareas de programación, como así también en la práctica y utilización de herramientas de soporte a la tarea de desarrollo de *software*.

De acuerdo a la estructura curricular este espacio dispone de 6 (seis) horas cátedra semanales, totalizando 16 semanas.

Se aplica un formato de aula taller, para abordar conceptos que básicamente persiguen el objetivo de incrementar en el alumno su habilidad para resolver problemas relacionados con los temas de la asignatura.

Se plantea un espacio para discutir, aplicar y ampliar conceptos de programación orientada a objetos abordados en la asignatura paralela Programación II. Por otro lado, se intenta modelar un ambiente creativo de niveles crecientes de realismo respecto del mundo laboral.

2) OBJETIVOS DE LA ASIGNATURA

- Que el alumno conozca entornos integrados de programación.
- Que el alumno conozca y aplique herramientas complementarias para el desarrollo, documentación y prueba de componentes de software.
- Que el alumno conozca las diferentes modalidades de programación, de acuerdo a la naturaleza del problema planteado, y los recursos tecnológicos existentes.
- Que se logre una integración vertical y horizontal con materias afines.
- Que se promueva la participación del alumno en trabajos de investigación y desarrollo.



Carrera:	Tecnatura Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

3) CONTENIDOS MÍNIMOS

- Edición y depuración de programas.
- Ejecución de programas codificados en C y Java.
- Funciones y archivos.
- Paradigmas de programación.
- Pruebas del software. Introducción a Calidad de Software.
- Documentación de programación.

4) METODOLOGÍA DE ENSEÑANZA

Se considera que cada clase es teórico-práctica, donde se aplican en forma inmediata los conceptos expresados teóricamente, por lo que el trabajo del alumno, en general, responderá al de un aula-taller.

Las clases teóricas se presentan sobre pizarra y/o elementos multimediales con presentación de ejemplos de aplicación con distinto nivel de completitud y complejidad, según corresponda. Estas clases se deben complementar con lecturas sugeridas por el docente de la bibliografía propuesta, o a través de investigación sobre artículos relacionados con el tema en cuestión.

Se plantean trabajos prácticos, casos de estudio, investigación y de extensión. Los trabajos están planteados con el objetivo de proveer al alumno actividades de ejercitación y fijación de conceptos. Su desarrollo y presentación son de carácter obligatorio para la regularización de la materia, pudiendo ser llevados adelante de manera individual o grupal.

El seguimiento de los alumnos se realizará en base al registro de su asistencia, evaluación y participación en clase.

En caso de quedar algún tema sin dictar por razones de fuerza mayor, el docente recomendará la bibliografía adecuada y material específico de lectura.

5) METODOLOGÍA DE EVALUACIÓN, INSTANCIAS DE RECUPERACIÓN Y RÉGIMEN DE PROMOCIÓN.

El alumno debe contar con al menos el mínimo de asistencias establecido por las resoluciones y ordenanzas vigentes. Se considerará media falta cuando el alumno se presente a clase con una tardanza de hasta 15 minutos transcurridos desde la toma de asistencia por parte del docente. Tardanzas mayores a 15 minutos, o un tiempo de ausencia considerable del alumno a clase se considerará como falta completa.

La entrega de trabajos prácticos para su evaluación será selectiva. En cada clase práctica serán evaluados diferentes alumnos, a través de la presentación de su carpeta de trabajos prácticos. (El formato de carpeta de trabajos prácticos se describe en el Anexo I). No se aceptarán copias de trabajos prácticos de otros grupos o integrantes o cualquier otra situación que determine un plagio de contenidos.



Carrera:	Tecnatura Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

En el caso de la elaboración de codificación de programas, es el alumno quien debe lograr la habilidad de obtener un *software* funcional y verificar la solución a partir del producto final obtenido. El profesor en este caso cumple un rol consultor y orientador en la guía del alumno.

La evaluación principal se prevé a través de **dos exámenes parciales** y un **examen global integrador (optativo para alumnos en condiciones obtener la aprobación directa)**, abordando los contenidos conceptuales y prácticos, con una única instancia de recuperación para cada uno de ellos. Los exámenes parciales se aprobarán con el porcentaje indicado por ordenanzas vigentes en el presente ciclo lectivo, es decir: **nota mayor o igual a 4 (cuatro)**.

Todas las evaluaciones deben dejar una constancia documental, es decir ser almacenadas en un archivo o escritas en papel. Esta evaluación (o una copia) es devuelta al alumno con las correcciones pertinentes. La cátedra archivará los exámenes una vez que los alumnos hayan analizado las correcciones respectivas, para que la misma pueda ser consultada durante el presente año lectivo. Las evaluaciones realizadas a través de la plataforma de educación a distancia contarán con la correspondiente devolución a través del mismo medio de evaluación.

El criterio de evaluación a seguir será presentado en la primera clase a efectos de que el alumno conozca tanto la estructura del mismo y su alcance, como así también el nivel pretendido para la regularización de la materia.

Para adquirir la condición de **Aprobación No Directa (AND)** el alumno debe:

- Cumplir los requisitos establecidos por el plan de estudio, resoluciones y ordenanzas suplementarias.
- Obtener el porcentaje de aprobación en cada una de las evaluaciones, o en la correspondiente evaluación de recuperación, esto es: **una nota igual o mayor a 4 (cuatro)**.
- Presentar la carpeta de trabajos prácticos visada de acuerdo a lo solicitado por el docente.

Una vez **aprobada (no directamente)** la asignatura, el alumno deberá rendir un examen final para su aprobación. El examen final se rinde a programa completo (esto es: el profesor puede indagar sobre cualquier punto del programa), exigiéndose el nivel técnico correspondiente.

El alumno para poder acceder a rendir el examen final deberá presentar la carpeta de trabajos prácticos, programa y planificación de la asignatura y documentación que acredite su identidad. En caso de no contar con alguno de estos elementos no podrá acceder al examen final.

Existe la posibilidad de **Aprobación Directa (AD)** de acuerdo a los siguientes requisitos:

- Cumplir con los requisitos de **Aprobación No Directa** de la asignatura listados anteriormente.
- Contar con una nota **superior o igual a 6 (seis)** en cada instancia de evaluación o en la recuperación de la misma. Esta situación permite acceder al examen Global Integrador. La nota del recuperatorio reemplazará la nota de la evaluación original, y será la definitiva para la instancia evaluada.
- Aprobar el examen **Global Integrador**, o recuperatorio del mismo con una nota igual o superior a 6 (seis). Esta nota conformará la **nota final** de la materia.



Carrera:	Tecnicatura Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

6) Descripción Sintética de Unidades

La **Unidad 1** es introductoria y tiene por objeto introducir al alumno en el mundo del desarrollo de *software* desde una perspectiva industrial, donde el mismo pueda observar roles, responsabilidades y funciones de cada uno de los actores involucrados en una *software factory*. Se analiza la solución propuesta a la luz de la organización creada para dicho fin. En esta unidad se prestará especial atención al uso de un entorno integrado de programación, asistiendo al alumno en la instalación del *software* necesario para el desarrollo de *software*, su correcta configuración y observación de las herramientas principales provista por el mismo. El resto de las unidades temáticas tienen por objetivo que el alumno analice y utilice diferentes herramientas para la asistencia de la programación de sistemas, aplicando tecnologías propias del entorno integrado de programación, atendiendo a los temas teóricos presentados previamente.

La **Unidad 2** promueve a que el alumno adopte una actitud crítica frente a los diferentes paradigmas de programación a la hora de proponer la solución a un problema. Se considera necesario que el alumno conozca las diferentes modalidades de programación frente a la diversidad de problemas computacionales que se puedan presentar. Esta unidad ha sido incluida con el objeto de ubicar al alumno en una situación donde pueda reconocer el contexto de un problema, la mejor solución tecnológica para resolverlo y el paradigma de programación adecuado.

La **Unidad 3** y tiene por objeto que el alumno relacione el algoritmo que dará solución al problema tanto con los datos involucrados en la misma, como así también con la tecnología que permite la persistencia de los mismos. Para ello se provee al alumno de documentación sobre el procesamiento de ficheros en lenguaje C y Java.

En la **unidad 4** se analizan diferentes IDEs a la luz de la estructura de diferentes características estructurales de proyectos de *software*. Se hace especial hincapié en proyectos de *software* en Java.

En la **unidad 5** se instruye al alumno en la modalidad de programación colaborativa, a través de la configuración de un grupo de trabajo. Se utiliza un entorno de versionado para llevar a cabo la práctica. Para ello se cuenta con un servidor configurado a tal fin.

En la **unidad 6** se presentan las principales pruebas que se realizan en el ciclo de vida de un sistema, haciendo el foco en la prueba unitaria. Para llevar a cabo tales pruebas se utiliza el producto JUnit para realizar pruebas unitarias en Java, y se mencionan otros productos.

En la **unidad 7** se abordan las pruebas de caja blanca a través de la utilización del *debugger* provisto por el IDE Netbeans.

En la **unidad 8** y **unidad 9** se analiza la máquina virtual de Java desde una perspectiva técnica, con el fin de que el alumno pueda plantear estudios y análisis de performance de los programas que realiza.

7) RECURSOS DIDÁCTICOS A UTILIZAR COMO APOYO A LA ENSEÑANZA.

Para utilizar la estrategia pedagógica anteriormente explicitada es necesario contar con recursos didácticos específicos:

- Laboratorio y PCs:



Carrera:	Tecnatura Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

Se estima que toda, o la mayor parte de cada clase, se dicte en un laboratorio de computadoras, intentando distribuir equitativamente las PCs disponibles y el tiempo de uso de cada una.

- **Software:**
Se estima utilizar un paquete de ofimática, entornos de desarrollo y herramientas de compilación y depuración habituales para los lenguajes C/C++ y Java. Ambos disponibles para plataformas Windows y Linux.
- **Recursos tradicionales:**
Marcadores y pizarra.
- Libros impresos o en formato electrónico
- Plataforma de educación a distancia.

8) ARTICULACIÓN HORIZONTAL Y VERTICAL CON OTRAS MATERIAS

La materia Laboratorio de Computación II se relaciona directamente con la materia Programación II, dictada en el mismo semestre de la carrera. El fin de la primera es proveer de mecanismos y metodologías para la realización de tareas de programación en forma más adecuada y precisa.

Para más detalle ver Plan curricular.



Carrera: Tecnicatura Superior en Programación.
 Cátedra: **LABORATORIO DE COMPUTACION II**
 Nivel: Primer año. Año lectivo: 2019.
 Semestre de cursado: Segundo Cantidad de horas semanales: 6
 Profesor Titular: **MONETTI, Julio**

9) CRONOGRAMA DE CLASES, ACTIVIDADES Y EVALUACIONES

Teoría	Práctica	Actividad Extra-Áulica
SEMANA 1		
<p>Introducción a la Materia</p> <p>LABORATORIO DE SOFTWARE FACTORY</p> <p>1.1 Concepto. Roles dentro de una SF. Integración del programador en el proceso de desarrollo de <i>software</i>.</p> <p>1.2 Proyecto de <i>software</i>. Estudio de Factibilidad.</p> <p>1.3 Actividades en una SF: Análisis y Diseño, Edición, Ejecución y Depuración. <i>Testing</i>. Versionado. Documentación. Otras actividades.</p>	<p>Práctico 1. Identificación de una tecnología adecuada para un conjunto dado de problemas. Entrega semana 2</p>	<p>* Creación de Grupos * Inscripción en la plataforma de Educación a Distancia</p> <p>Evaluación Diagnóstica. Se evalúan aspectos básicos de programación, y temas preliminares de programación orientada a objetos</p>
SEMANA 2		
<p>PARADIGMAS DE PROGRAMACIÓN</p> <p>2.1 Concepto de Paradigma de Programación.</p> <p>2.2 Características del Paradigma Orientado a Objetos. Estructura de un programa OO.</p> <p>2.3 Características del Paradigma Funcional: Estructura de un programa funcional</p> <p>2.4 Características del Paradigma Lógico: Estructura de un programa lógico.</p> <p>2.5 Programación paralela: concepto. Ambientes de desarrollo. Conceptos básicos de programación concurrente.</p>	<p>* Discusión sobre: Paradigmas de Programación. Tipos de software (base, aplicación, servicios). Alto y bajo nivel. Herramientas de apoyo a la programación.</p> <p>Práctico 2. Paradigmas de Programación. Entrega semana 3.</p>	<p>Autoevaluación sobre la Plataforma de Educación a Distancia. Tema: Paradigmas de Programación</p>
SEMANA 3		



Carrera: Tecnicatura Superior en Programación.
Cátedra: **LABORATORIO DE COMPUTACION II**
Nivel: Primer año. **Año lectivo:** 2019.
Semestre de cursado: Segundo **Cantidad de horas semanales:** 6
Profesor Titular: **MONETTI, Julio**

ALMACENAMIENTO DE DATOS 3.1 El fichero de datos como complemento a la aplicación ejecutable. 3.2 Organización de ficheros. 3.3 Introducción a la seguridad sobre los ficheros de datos. Bloqueos y permisos. 3.4 Introducción a las bases de datos. La base de datos como complemento a la aplicación ejecutable. 3.5 Otras fuentes de datos: <i>sockets</i> y llamadas a servicios.	Práctico Almacenamiento de Datos. Entrega semana 4	Lectura material de estudio
SEMANA 4		
EDICIÓN Y EJECUCIÓN DE PROGRAMAS 4.1 Edición de código fuente. IDEs más comunes (de consola y gráfico). Características de los principales IDEs. 4.2 Ejemplos de utilización. 4.3 Administración de un proyecto de <i>software</i> Java. 4.4 Mejoramiento del código: <i>Refactoring</i> . Ejemplos y utilización del refactorizador. 4.5 Proyectos Java-Maven.	Práctico 4. Edición de programas. Entrega semana 5	
SEMANA 5		
VERSIONADO 5.1 Conceptos Básicos de Versionado. 5.2 <i>Subversion</i> SVN. 5.3 Configuración de un proyecto con <i>Subversion</i> SVN. 5.4 Otras herramientas de versionado para trabajo colaborativo en proyectos de <i>software</i> . 5.5 Comandos y operaciones básicas. Conflictos.	Práctico 5. Versionado. Entrega semana 6	Autoevaluación sobre la Plataforma de Educación a Distancia. Tema: Versionado
SEMANA 6		



Carrera: Tecnicatura Superior en Programación.
Cátedra: **LABORATORIO DE COMPUTACION II**
Nivel: Primer año. Año lectivo: 2019.
Semestre de cursado: Segundo Cantidad de horas semanales: 6
Profesor Titular: **MONETTI, Julio**

Práctica Integradora de Programación		
SEMANA 7		
PRUEBAS DE SOFTWARE 6.1 Prueba de <i>Software</i> : Fallos, defectos y errores. Tipos de defectos. Conceptos técnicos y de uso. Alcance de una prueba de <i>software</i> . 6.2 Prueba Unitaria. Prueba de Integración. Prueba de validación y aceptación. Prueba del sistema. 6.3 Diseño de prueba de escritorio. Diseño de archivos de bitácora. 6.4 Técnicas y Estrategias para la prueba de <i>software</i> . 6.5 Organización para las pruebas de <i>software</i> . Verificación y validación de programas. 6.6 Uso de JUnit: clases de pruebas. Aserciones. Métodos de la clase de prueba.	Práctico 6. Pruebas de <i>software</i> . Entrega semana 8	Autoevaluación sobre la Plataforma de Educación a Distancia. Tema: Testing
SEMANA 8		
DEPURACIÓN DE PROGRAMAS 7.1 Depuración (<i>debugging</i>): Conceptos. 7.2 Herramientas de depuración: Trazas y Aserciones, Depuradores. 7.3 Depuradores de memoria, seguidores de llamadas, 7.4 Tipos de <i>breakpoints</i> , <i>step-over</i> , <i>step-into</i> , <i>step-out</i> , <i>move-to-cursor</i> , 7.6 Inspección de variables. 7.6 Otra información provista por <i>debuggers</i> . 7.7 Uso de Log4J.	Práctico 7. Depuración de Programas. Entrega: semana 9	
SEMANA 9		
PARCIAL 1		



Carrera: Tecnicatura Superior en Programación.
Cátedra: **LABORATORIO DE COMPUTACION II**
Nivel: Primer año. **Año lectivo:** 2019.
Semestre de cursado: Segundo **Cantidad de horas semanales:** 6
Profesor Titular: **MONETTI, Julio**

SEMANA 10		
ENTORNOS DE EJECUCIÓN JAVA 8.1 Programas ejecutables: <i>standalone</i> , <i>GUI</i> , <i>librería</i> . 8.2 Concepto de arquitectura cliente-servidor. 8.3 La Máquina Virtual de Java: Su uso: y posibilidades de configuración. 8.4 Formas alternativas de ejecución: Los <i>Applets</i> y su entorno de ejecución. 8.5 <i>WebServices</i> : entorno de trabajo y consumo de servicios. <i>Servlets</i> y su entorno de ejecución. 8.6 Performance. Uso del <i>profiler</i> para el análisis de performance.	Práctico 8. Entornos de Ejecución de Programas JAVA. Entrega semana 11	Autoevaluación sobre la Plataforma de Educación a Distancia. Tema: Entornos de Ejecución Java
SEMANA 11		
LA MÁQUINA VIRTUAL DE JAVA 9.1 Arquitectura. Modelo de Memoria. 9.2 Uso de la memoria. Proceso de recolección de basura. 9.3 Uso de la JVM y posibilidades de configuración. Tasas de permanencia, localización y promoción.		Lectura de material de estudio
RECUPERATORIO PARCIAL 1		
SEMANA 12		
9.4 <i>Tuning</i> de la máquina virtual de Java (JVM). 9.5 Análisis de procesos de la máquina virtual de Java. 9.6 Administración de hilos de ejecución. 9.7 <i>Profiling</i> . Posibilidades de perfilado en Netbeans.	Práctico 9. <i>Profiling</i> . Entrega semana 11	
SEMANA 13		



Carrera: Tecnicatura Superior en Programación.
Cátedra: LABORATORIO DE COMPUTACION II
Nivel: Primer año. **Año lectivo:** 2019.
Semestre de cursado: Segundo **Cantidad de horas semanales:** 6
Profesor Titular: MONETTI, Julio

<p>INTRODUCCIÓN A LA CALIDAD DE SOFTWARE Y DOCUMENTACION EN PROGRAMACIÓN</p> <p>10.1 Definición de calidad, fiabilidad de software. 10.2 Relación de la ingeniería de <i>software</i> con el proceso de desarrollo. 10.3 Métodos y Herramientas. Normas y estándares. 10.4 Introducción a las métricas de programación. 10.5 Métricas para el código fuente y diseño de la aplicación.</p> <p>10.6 Documentación de procesos. Documentación de base de datos. 10.7 Diseño de manuales de programación: destinatarios, componentes. 10.8 Diseño del manual del usuario.</p>	<p>Práctico 10. Calidad de Software. Entrega: semana 14</p>	<p>Autoevaluación sobre la Plataforma de Educación a Distancia. Tema: Métricas Técnicas Orientadas a Objetos</p>
<p>Parcial 2</p>		<p>Presentación de la plataforma Sonarqube.</p>
SEMANA 14		
<p>RECUPERATORIO PARCIAL 2</p>		
SEMANA 15		
<p>Global Integrador (Examen escrito)</p>		
SEMANA 16		
<p>RECUPERATORIO GLOBAL INTEGRADOR</p>		



Carrera: Tecnicatura Superior en Programación.
Cátedra: **LABORATORIO DE COMPUTACION II**
Nivel: Primer año. Año lectivo: 2019.
Semestre de cursado: Segundo Cantidad de horas semanales: 6
Profesor Titular: **MONETTI, Julio**

10) BIBLIOGRAFÍA, GUÍAS DE ESTUDIO, PUBLICACIONES VINCULADAS CON LA MATERIA.

- Medición y Estimación de Software: Técnicas y Métodos para Mejorar la Calidad y la Productividad. Mario Piattini Velthui y otros. Alfaomega. 2008.
- Pro Netbeans 6. Rich Client Platform Edition. Adam Myatt. Apress.
- Ingeniería de Software, un enfoque práctico. Roger Pressman. 6ta edición.
- Linux Desmitificado. Manuel Alberto Ricart.
- Essential System Administration. Eleen Frisch.
- Java 2: Lenguajes y Aplicaciones. F. J. Ceballos.

11) LECTURAS COMPLEMENTARIAS

- Conceptos de Cómputo Paralelo. José Torres Jiménez & Eduardo Arturo Rodríguez Tello.
- Guías de estudio elaboradas por la cátedra.
- Java 2. Manual del Usuario y Tutorial. Agustín Froufe.
- Cómo programar en JAVA. Deitel & Deitel.
- Integración Linux-Windows. Mike Mc Cune.
- Fundamentos de Programación. Piensa en C. Osvaldo Cairó.
- Ingeniería de Software Orientada a Objetos con UML, Java E Internet. Alfredo Weitzenfeld.
- Programación Orientada a Objetos. Técnicas Avanzadas de Programación. Carlos Fontela.

Lugar y fecha: Mendoza, Febrero de 2019	PROFESOR/ES A CARGO Apellido y nombre: Julio C. MONETTI N° de Legajo: 35459
	Firma:



Carrera:	Técnica Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

ANEXO I. Formato de la Carpeta de Trabajos Prácticos y Especificaciones para su Aprobación.

La carpeta de trabajos prácticos deberá seguir los siguientes lineamientos:

1. Tamaño de carpeta y folios A4: 21cm x 29,7cm.
2. Todas las hojas del práctico deben contener número de folio, excepto la carátula.
3. Márgenes
 - a. Izquierdo y derecho 3cm
 - b. Superior e inferior 1.5cm
4. Carátula especificando
 - a. Nombre de la asignatura
 - b. Año de cursado y turno.
 - c. Nombre del alumno y profesor.
 - d. Nombre de los integrantes del Grupo.
5. Durante el cursado se exigirá **una copia** de la carpeta por grupo para su visado.
6. Cada alumno del grupo deberá presentar su propia copia al final del semestre.
7. Copia de trabajos de otros grupos serán penalizados y desaprobados.
8. Texto de páginas WEB, libros u otras publicaciones deberá ser debidamente citado o será considerado como plagio al autor original, dando posibilidad a la desaprobación del práctico.

Contenido de los Trabajos Prácticos

Los trabajos prácticos tienen por objetivo ampliar los conocimientos adquiridos en las clases teóricas. Cada práctico está compuesto por dos partes:

1. **Parte A.** Ampliación de conceptos a través de lecturas sugeridas.
2. **Parte B.** Desarrollos prácticos. Corresponde al desarrollo de algoritmos, su prueba, verificación, análisis de eficiencia, etc, y tareas similares donde el alumno pueda exponer la pericia adquirida sobre el tema en cuestión.



Carrera:	Tecnatura Superior en Programación.	
Cátedra:	LABORATORIO DE COMPUTACION II	
Nivel:	Primer año.	Año lectivo: 2019.
Semestre de cursado:	Segundo	Cantidad de horas semanales: 6
Profesor Titular:	MONETTI, Julio	

CONOCIMIENTOS PREVIOS PARA EL CURSADO DE LA MATERIA

Para el cursado de la presente materia es necesario que el estudiante domine los temas dictados en las materias Laboratorio de Computación I y Programación I.

Que posea las destrezas necesarias para la codificación de algoritmos básicos, como por ejemplo algoritmos de ordenamiento.

Que sepa reconocer y utilizar estructuras de datos, como por ejemplo matrices n-dimensionales.

Que conozca en general las herramientas necesarias para la construcción de programas.